

# Uvod u grafički korisnički interfejs

Grafički korisnički interfejs (GUI – graphical user interface) predstavlja interfejs kroz koji korisnik može upravljati željenim uređajem koristeći vizualne indikatore. U ovom kursu ćemo se baviti kreiranjem korisničkih interfejsa koristeći Tkinter, naučiti šta je platno i kako da kreiramo elemente na njemu, kao i baviti se animacijom. Pored animacije, saznaćemo kako funkcionišu ulazno-izlazni tokovi u aplikaciji, ali i kako izvršiti detekciju sudara.

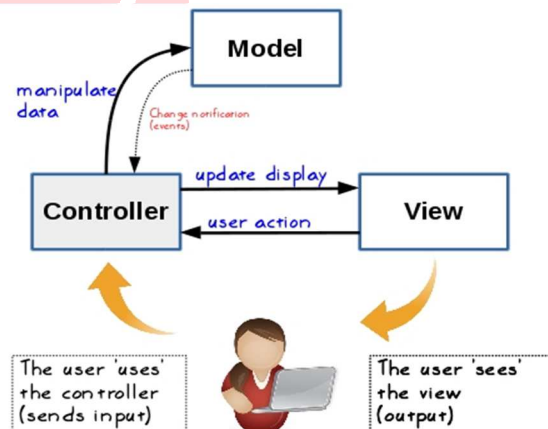
## Šta je grafički korisnički interfejs?

Grafički korisnički interfejs je razvijen u kasnim 1970-im godinama u američkoj Xerox laboratoriji za razvoj, a komercijalni uspeh je doživeo u Appleovom Macintosh i Microsoftovom Windows operativnom sistemu. Razloga za razvoj ovakvog pristupa bilo je mnogo, a najbitniji među njima je neefikasno korišćenje interfejsa baziranih na komandnim linijama.

Nakon tog trenutka, grafički interfejsi postaju standard u aplikativnom programiranju, čime se krajnjem korisniku pruža mogućnost intuitivnog upravljanja računarom i drugim uređajima kroz direktnu manipulaciju dugmićima, vertikalnim scroll trakama itd. U poslednje vreme prisutan je trend grafičkih korisničkih interfejsa gde korisnik tim elementima upravlja posredstvom ekrana na dodir, odnosno touchscreen tehnologije, kao i trend korišćenja glasovnih komandi.

## Kako grafički korisnički interfejs funkcioniše?

Principi grafičkog korisničkog interfejsa se baziraju na MVC (model-view-controller) šablonu, koji razdvaja podatke od načina na koji se oni prikazuju korisniku, iz čega proizilazi i koncept gde se korisniku prikazuju koje funkcionalnosti sa tim podacima su moguće, umesto prethodnog rešenja, gde se od korisnika tražilo kućanje komandi u komandnom okruženju kako bi se manipuliralo tim podacima. Na ovaj način, korisnici manipulišu podacima kroz vidžete (widgets – elementi grafičkog interfejsa kao što su dugmad, liste, kontrole za unos podataka itd.), koji su dizajnirani za rad sa različitim tipovima podataka.



Slika 1.1. Šematski prikaz funkcionisanja MVC modela<sup>1</sup>

<sup>1</sup> [https://subscription.packtpub.com/book/application\\_development/9781785889196/10/ch10lvl1sec88/mvc-architecture](https://subscription.packtpub.com/book/application_development/9781785889196/10/ch10lvl1sec88/mvc-architecture)

Izgled same aplikacije može uvek lako biti redizajniran po volji korisnika, jer je nezavistan od samih podataka koje ta aplikacija poseduje. GUI aplikacije uglavnom implementiraju sopstvene ideje za dizajn interfejsa, iako sam operativni sistem nudi već ugrađena, slična rešenja. Ovo se najpre odnosi na izgled prozora, formi, dugmića, lista itd.

## Prednosti i mane grafičkog korisničkog interfejsa

Kao i svako rešenje u programiranju, tako i upotreba grafičkog interfejsa, koja dosta toga olakšava, ima i prednosti i mane.

### Prednosti:

- jednostavnost korišćenja;
- intuitivan pristup upravljanja programom;
- značajno jednostavnija pretraga podataka;
- prednost se ogleda i u jednostavnosti obavljanja traženih funkcija, pošto je prethodni način – izvršavanje komandi preko komandne linije – zahtevao i bar minimalno poznavanje programiranja od krajnjeg korisnika.

### Mane:

- krajnji korisnik može koristiti samo funkcije koje je programer ugradio u grafički interfejs;
- aplikacija sa grafičkim interfejsom zauzima više resursa računara nego aplikacija koja funkcioniše kroz komandnu liniju;
- pored veće zauzetosti resursa, grafičke aplikacije su sporije kada je reč o izvršavanju zadataka u poređenju sa aplikacijama koje su bazirane na komandnoj liniji;
- grafički interfejsi su krajnjem korisniku najlakši način korišćenja programa, ali za programere koji ih kreiraju važi suprotno, jer pored svih funkcionalnosti, programer mora razmišljati i o tome na koji način kreirati grafički interfejs koji bi se kao viši, zaseban sloj nadovezao na tu funkcionalnost i bio intuitivan krajnjem korisniku.

## Primeri GUI aplikacija i interfejsa

Za prvu GUI aplikaciju se smatra program Sketchpad, koji je 1962. godine razvio Ivan Sutherland sa MIT-a. Ova aplikacija se sastojala od olovke na bazi svetla pomoću koje su korisnici mogli da upravljaju objektima na ekranu u realnom vremenu. Ovaj program se smatra i pretečom CAD (computer aided design) programa i osnovom za dalji razvoj objektno orijentisanog programiranja.



Slika 1.2. Ivan Sutherland koristi Sketchpad<sup>2</sup>

<sup>2</sup> <https://bimaplus.org/news/the-very-beginning-of-the-digital-representation-ivan-sutherland-sketchpad/>

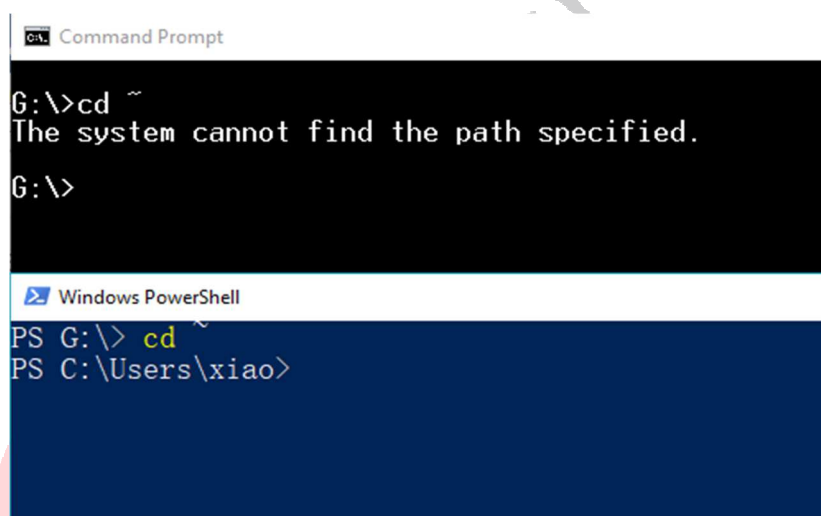
Moderni operativni sistemi i grafički interfejsi su umetnuti u skoro svaku interaktivnu aplikaciju kao što su bankomati, pametni telefoni, video-igre. Neki od poznatijih grafičkih interfejsa su:

- za desktop uređaje: Microsoft Windows, macOS, Ubuntu Unity, GNOME Shell;
- za mobilne uređaje: Android, iOS, BlackBerry OS, Windows 10 Mobile, Firefox OS.

## Tipovi korisničkih interfejsa

Do sada je bilo reči uglavnom o grafičkim korisničkim interfejsima, a postoje i drugi korisnički interfejsi sa kojima smo se susreli: komandni korisnički interfejsi i web korisnički interfejsi.

Komandni korisnički interfejsi su zapravo komandne linije koje funkcionišu tako što korisnik putem tastature piše komande i sistem ih izvršava. Ovi interfejsi podržavaju skriptovanje, kao i automatizaciju, što nam omogućava viši i fleksibilniji nivo funkcionalnosti u odnosu na grafičke interfejse. Bili su široko rasprostranjeni do pojave grafičkih korisničkih interfejsa. Neki od popularnih komandnih interfejsa su Command Prompt i PowerShell u Windowsu, kao i Terminal na UNIX sistemima.



*Slika 1.3. Command Prompt i PowerShell<sup>3</sup>*

Sa druge strane, imamo web korisnički interfejs (browser user interface – BUI), gde korisnik upravlja softverom koji se nalazi na serveru, dok mu interfejs predstavljaju pregledač i ta web stranica. Prednost ovakvih interfejsa je to što su nezavisni od platforme – mogu se pokretati na bilo kojoj platformi koja ima pristup internetu i podržava upotrebu browsera (pregledača). Prednost web korisničkih interfejsa je i to što se lako održavaju i zahtevaju manje resursa od korisnika – jedino što je potrebno za njihovo funkcionisanje je zapravo internet konekcija.

<sup>3</sup> <https://revistarai.org/How-to/CMD-vs-Windows-PowerShell-on-Windows-10>



Slika 1.4. Izgled web korisničkog interfejsa za konfigurisanje WiFi rutera<sup>4</sup>

## Pitanje

Za ime koje aplikacije se vezuje nastanak GUI aplikacija?

- SketchNote
- OneNote
- **Sketchpad**

## Objašnjenje:

*Tačan odgovor je program Sketchpad, razvijen 1962, za čije postojanje se vezuje začetak GUI aplikacija.*

## Elementi grafičkih korisničkih interfejsa

Grafički interfejsi se sastoje od elemenata. Iako postoje grafički korisnički interfejsi kojima korisnik upravlja na dodir (touch interface) i slično, u ovoj nastavnoj jedinici ćemo se fokusirati na desktop GUI aplikacije i njihove elemente. Tako kreirane desktop aplikacije prate WIMP paradigmu, što je zapravo skraćenica od: *window, icon, menus, pointer*, o čemu se više može pročitati na ovom [linku](https://en.wikipedia.org/wiki/Browser_user_interface). Na sličnim principima funkcioniše [post-WIMP](#).

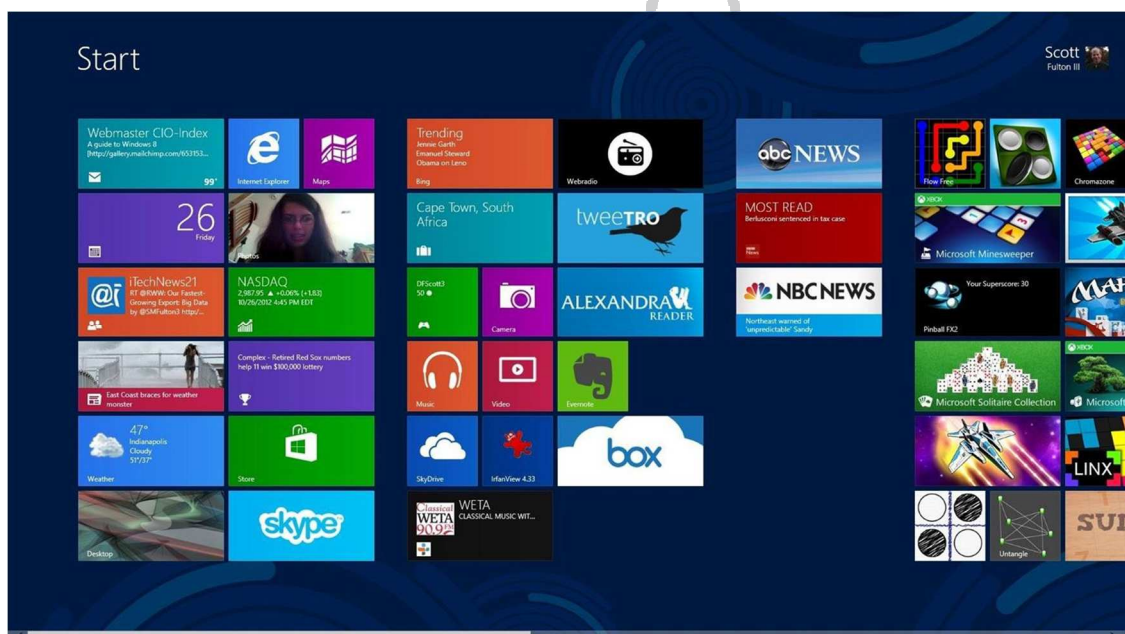
<sup>4</sup> [https://en.wikipedia.org/wiki/Browser\\_user\\_interface](https://en.wikipedia.org/wiki/Browser_user_interface)

WIMP je popularizovao Apple sa svojim Mekintošom (Macintosh) iz 1984. godine, koji je predstavio koncept linija menija i dodatno razvio dotadašnje upravljanje prozorima. Jedan takav WIMP interfejs se sastoji od sledećih komponenata:

- prozor, koji se pokreće u zasebnom programu koji je izolovan od drugih pokrenutih programa na računaru;
- ikonica, koja predstavlja prečicu ka određenoj akciji koju računar izvršava (primer: prečica koja predstavlja putanju do željenog programa);
- meni predstavlja sistem selekcije baziran na tekstu ili na slikama (ikonicama) i tekstu, kojim se mogu izvršavati određene komande; funkcionalnosti menija zavise od konteksta u kojem se koriste;
- pokazivač je simbol koji na ekranu predstavlja pokret fizičkog uređaja kojim korisnik upravlja kako bi kontrolisao, obeležavao i izvršavao naredbe itd.

Primeri WIMP aplikacija koje se i danas koriste su, recimo, Microsoft Word, Microsoft Excel, Google Chrome...

Post-WIMP način kreiranja aplikacija je nastao kada se javila potreba za specifičnijim aplikacijama, kao što su, na primer, video-igre (kod igračkih konzola ponajviše), bankomati, MP3 plejeri...



Slika 1.5. Izgled post-WIMP koncepta primenjen na Windows 8 start meniju<sup>5</sup>

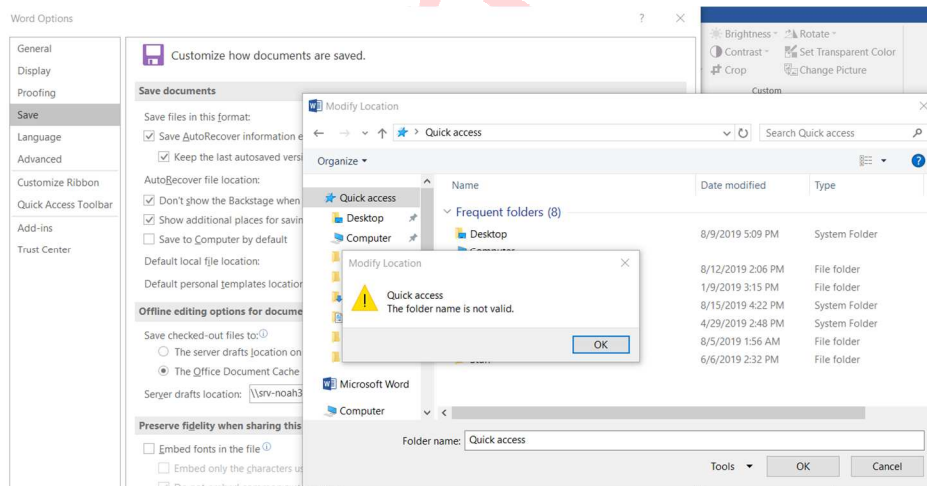
Bilo da je reč o WIMP stilu ili post-WIMP stilu, koncept je dosta sličan. Elementi WIMP stila GUI aplikacije se mogu podeliti u osam kategorija:

<sup>5</sup> <https://windows.gadgetsacks.com/how-to/bring-classic-start-menu-back-windows-8-0139983/>

## Prozor (window)

Prozor predstavlja segment ekrana koji prikazuje informacije. Njegov sadržaj je prikazan nezavisno od ostatka ekrana. Prozore smo zasigurno do sada već videli u praksi. Svaki put kada otvorimo aplikaciju – ona se otvori ili u svom prozoru ili u prozoru komandne linije. Isto tako, otvaranje bilo kog foldera se vrši u prozoru File Explorera. Korisnik jednostavno upravlja prozorima tako što ih može pomerati, smanjivati/povećavati i slično. U modernim operativnim sistemima, prozori aplikacija se mogu postavljati jedan iza/ispred drugog. Takođe, u tim istim sistemima je moguće i otvaranje više prozora odjednom, gde je sistemska radna memorija jedini limit kada je reč o broju prozora koji mogu biti otvoreni. Kako je upotreba prozora u operativnim sistemima rasla, pojavila se potreba za njihovom kategorizacijom i specijalizacijom na sledeći način:

- kontejnerski prozori – vrsta prozora koja se otvara npr. nakon otvaranja sadržaja direktorijuma koji se sastoji od liste drugih ikonica/fajlova/direktorijuma i sl.;
- pretraživački prozori – omogućavaju korisnicima da se „kreću“ napred ili nazad od trenutne lokacije; ovde je reč najčešće o internet pregledačima, ali postoje i druge aplikacije koje nude istu funkcionalnost, a omogućavaju nam čitanje dokumenata, listanje slika u galerijama itd.;
- tekstualni terminali – prozori koji su dizajnirani za povezivanje sa tekstualnim interfejsima – najpre komandnim linijama (komandni korisnički interfejs), a koji mogu biti deo grafičkih aplikacija;
- izvedeni prozori (child window) – prozori koji se otvaraju kao rezultat prethodne akcije korisnika nad glavnim prozorom; najbolji primer ovoga su pop-up prozori, koje često srećemo u internet pregledačima, ali nisu samo njima svojstveni;



Slika 1.6. Prikaz izvedenog prozora<sup>6</sup>

- prozori sa porukama – prozori sa porukama ili dijalog prozori se mogu kategorisati i kao izvedeni prozori, ali se zbog svoje uske funkcionalnosti nekada i zasebno kategorišu; reč je o prozorima male veličine koje otvara sam program kako bi nam prikazao određenu informaciju; ovakav prozor uglavnom sadrži jedno, a ređe dva dugmeta i korisnik je u obavezi da klikom na neki od tih dugmića isključi taj prozor.

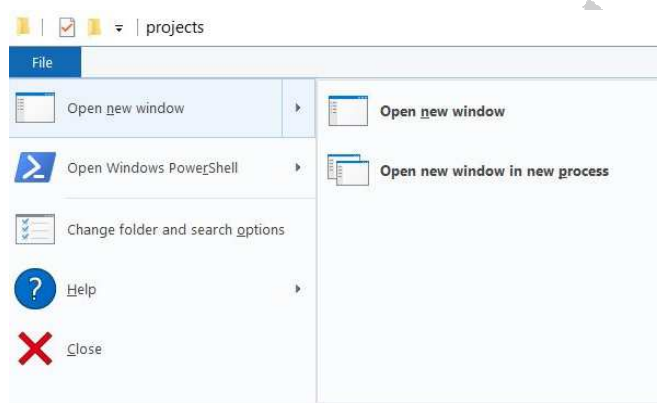
<sup>6</sup> <https://superuser.com/questions/1474202/set-quick-access-as-default-save-location-in-ms-office>



## Meniji

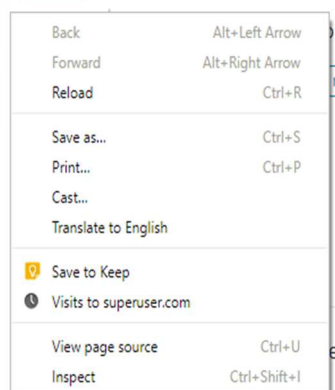
Meniji omogućavaju korisniku izvršavanje komandi koje se nalaze u listi izbora. Opcije se mogu izabrati mišem, prečicama na tastaturi ili nekim drugim uređajem koji prima korisnički ulaz. Meniji se mogu sastojati od sledećih komponenata:

- linija menija (menu bar) – prikazana je horizontalno pri vrhu samog prozora; padajući meniji se takođe često asociraju sa linijom menija, jer nakon što korisnik klikne na jednu opciju iz linije menija, otvoriće mu se padajući meni sa još opcija;
- meni je osnovna komponenta linije menija i sadrži svoje ime; može, ali ne mora sadržati pod-elemente; ako ih poseduje, oni se ukazuju korisniku nakon što klikne na njega ili pokazivačem miša pređe preko njega (padajući meni); klikom bilo gde drugde, izvan, ovaj padajući meni koji je deo menija nestaje.



*Slika 1.7. Prikaz linije menija sa podelementima*

- kontekstni meni – ovaj meni nije vidljiv korisniku sve dok korisnik ne izvrši određenu radnju mišem ili drugim pokazivačem, npr. desni klik miša; kada se ta određena, predefinisana radnja izvrši, ovaj meni se prikazuje korisniku.



*Slika 1.8 Primer kontekstnog menija internet pregledača aktiviranog desnim klikom<sup>7</sup>*

<sup>7</sup> <https://superuser.com/questions/1276983/right-click-goes-back-a-page>

## Ikonice

Ikonice su male slike koji predstavljaju objekte kao što su fajlovi, programi, dokumenti ili komande. Predstavljaju brz način korisniku za izvršavanje komandi: na primer, duplim klikom na ikonicu određenog dokumenta – otvorićemo taj dokument u podrazumevanom programu za rad sa njim. Ikonice su korisne i zato što će fajlovi istog tipa, u većini modernih operativnih sistema, imati iste ikonice, što olakšava vizualnu pretragu po direktorijumu.

## Kontrole (widgets)

Elementi koji se nalaze u samom interfejsu, u prozoru, nazivaju se još i kontrole – vidžeti. Vidžeti (widgets) su softverske komponente koje omogućavaju korisniku interakciju sa grafičkim interfejsom. Svaki vidžet ima svoju funkciju, čime se korisniku upotrebom različitih vidžeta pruža široku funkcionalnost. Sa vidžetima smo se svakako susreli ako smo koristili bilo koju GUI aplikaciju. Pojavljuju se kao meniji, dugmad, labele, polja za potvrdu, trake za pomeranje, radio-dugmići itd.

## Tabovi (kartice)

Tabovi su uglavnom pravougaoni okviri u grafičkom interfejsu koji se sastoje od imena i ikonice i obuhvataju određeni, tematski srodan grupisani sadržaj. Sa tabovima se često susrećemo u dijalozima Windows operativnog sistema, kao i u internet pregledačima.

## Kursor (pokazivač)

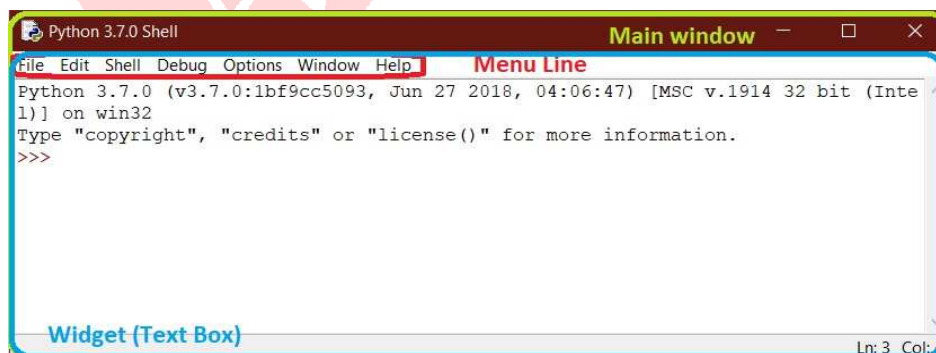
Kursor je indikator koji predstavlja poziciju na ekranu na čijem će se mestu izvršiti korisnička akcija koja se obavlja najčešće preko miša, laptop tačpeda ili dodira.

## Pokazivač mesta umetanja (insertion point)

Pokazivač mesta umetanja ili tekstualni kursor predstavlja mesto na korisničkom interfejsu gde je trenutni fokus. Predstavlja mesto i objekat nad kojim će se izvršiti naredna korisnička instrukcija, kao što je npr. pisanje teksta, selekcija, kopiranje –umetanje operacija itd.

## Selekcija

Selekcija je lista elemenata nad kojima će se izvršiti naredna korisnička akcija. Ovde se mogu naći dva scenarija: korisnik sam dodaje elemente u listu selekcije i računar, odnosno aplikacija, umesto korisnika selektuje elemente.



Slika 1.9 Elementi grafičkog interfejsa Python interpretera



## Rezime

- Grafički korisnički interfejs je razvijen kasnih 1970-ih godinama u Xerox Palo Alto laboratoriji za razvoj, a komercijalni uspeh je doživeo u Appleovom Mekintoš (Macintosh) i Microsoftovom Windows operativnom sistemu.
- Principi grafičkog korisničkog interfejsa se baziraju na MVC (model-view-controller) šablonu, koji razdvaja podatke od načina na koji se prikazuju korisniku, iz čega proizilazi i koncept gde se korisniku prikazuje koje funkcionalnosti sa tim podacima su moguće, umesto prethodnog rešenja, gde se od korisnika tražilo kucanje komandi u komandnom okruženju kako bi se manipuliralo tim podacima.
- Izgled same aplikacije može uvek lako biti redizajniran po volji korisnika, jer je nezavistan od samih podataka koje ta aplikacija poseduje. GUI aplikacije uglavnom implementiraju sopstvene ideje za dizajn interfejsa, iako sam operativni sistem nudi već ugrađena, slična rešenja. Ovo se najpre odnosi na izgled prozora, formi, dugmića, lista itd.
- Za prvu GUI aplikaciju se smatra program Sketchpad, koji je razvio Ivan Sutherland sa MIT-a 1962. godine; ovaj program podrazumevao je korišćenje olovke na bazi svetla, pomoću koje su korisnici mogli upravljati objektima na ekranu u realnom vremenu.
- Postoji više vrsta korisničkih interfejsa: komandni, web i grafički korisnički interfejsi.
- Elementi grafičkih interfejsa su prozori, meniji, ikonice, kontrole, tabovi, elementi interakcije, kursor, pokazivač mesta umetanja i selekcija.

